



THE QUEUING MANPOWER MODEL (QMAN)

Jeffrey H. Grobman, Capt, USAF
David M. Quick, 1Lt, USAF
William M. Weaver, Maj, USAF

HUMAN RESOURCES DIRECTORATE
MANPOWER AND PERSONNEL RESEARCH DIVISION
7909 Lindbergh Drive
Brooks AFB, Texas 78235-5352

19961106 162

February 1996

Final Technical Paper for Period September 1992-April 1994

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
BROOKS AIR FORCE BASE, TEXAS

NOTICES

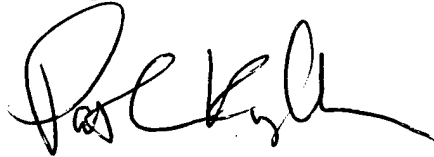
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or in any manner construed, as licensing the holder, or any other person or corporation: or as conveying any rights in any way to be related thereto.

The Office of Public Affairs has reviewed this paper and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nations.

This paper has been reviewed and is approved for publication.



JEFFREY H. GROBMAN, Capt, USAF
Project Scientist



PATRICK C. KYLLONEN
Technical Director
Manpower and Personnel Research Division



GARY D. ZANK, Col, USAF
Chief
Manpower and Personnel Research Division

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE February 1996	3. REPORT TYPE AND DATES COVERED Final Technical Paper - September 1992-December 1994		
4. TITLE AND SUBTITLE Queuing Manpower Model (QMAN)		5. FUNDING NUMBERS PE - 62205F PR - 7719 TA - 19 WU - 29		
6. AUTHOR(S) Jeffrey H. Grobman, Capt, USAF William M. Weaver, Maj, USAF David M. Quick, 1Lt, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Manpower and Personnel Research Division 7909 Lindbergh Drive Brooks AFB TX 78235-5352		8. PERFORMING ORGANIZATION REPORT NUMBER AL/HR-TP-1995-0021		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES Armstrong Laboratory Technical Monitor: Capt. Jenkins, 210-536-3551				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) The Queuing Manpower Model (QMAN) is an analytic personal computer (PC) based model for the determination of maintenance manpower requirements. The model applies a queuing algorithm to Air Force Specialty (AFS)/crew size clusters to determine necessary manning to meet flying demands. This value is then compared to work load and crew size manpower demands to determine actual requirements. The Turbo Pascal implementation of QMAN provides rapid manpower estimations that compare favorably with the Air Force standard maintenance manpower model, the Logistics Composite Model (LCOM). The inherent speed of an analytic model, such as QMAN, contrasts with the lengthy run-times required by large and complicated Monte Carlo simulations like LCOM. These lengthy run-times lead to lengthy analysis due to the need for multiple simulation runs before "optimal" manpower requirements can be determined. QMAN makes possible various types of analyses that were not previously feasible due to LCOM time demands. Examples of these include the determination of the effect of increased maintainer productivity, shorter flying days, changing wing structure, and alternative occupational structures on manpower requirements. Additionally, QMAN serves as a tool for assessing and reducing manpower costs for aircraft systems under development or modification.				
14. SUBJECT TERMS Logistics Composite Model Utilization rates Manpower estimation Queuing		15. NUMBER OF PAGES 60		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

INTRODUCTION	1
METHODS, ASSUMPTIONS, AND PROCEDURES	2
Statement of the Problem	2
Queuing Theory Application	2
Approach	3
Number of Personnel per AFS (X_i) as a Result of Queuing Demand	3
Number of Personnel per AFS (F_i) as a Result of Utilization Adjusted Workload Effects	6
Number of Personnel per AFS (G_i) as a Result of Required Maximum Crew Sizes	7
Assumptions of Model	7
IMPLEMENTATION	7
EVALUATION	9
EXAMPLES OF QMAN APPLICATIONS	10
Effect of an Increase in Maintainer Productivity on Manpower	10
Effect of Aircraft Reliability on Aircraft Maintainer Requirements	11
CONCLUSION	12
REFERENCES	13
APPENDIX A: QMAN SOURCE CODE	14
APPENDIX B: QMAN TO LCOM COMPARISON (F-22)	42
APPENDIX C: QMAN TO LCOM COMPARISON (F-15)	47

LIST OF TABLES

Table 1. Comparison of QMAN and LCOM Manpower Estimates	10
Table 2. Effect of Increase in Productivity on Number of Manpower Positions Required	11

LIST OF FIGURES

Figure 1. QMAN Input File	8
Figure 2. QMAN Output Screen	9
Figure 3. Effect of Aircraft Reliability on Manpower Requirements	12

PREFACE

This work is part of the Manpower and Personnel Research Division's in-house research program to develop an understanding of the relationship between maintenance manpower requirements and sortie generation, job structures, maintainer productivity, and aircraft reliability. Work was completed under the in-house exploratory research program - Development of Acquisition-related Manpower, Personnel, and Training (MPT) Analysis Methodologies (WU 1123A702).

The authors wish to thank Mr. Larry Looper and Dr. Jacobina Skinner for their valuable technical contribution to this effort. Completion of this technical paper would not have been possible without their participation and guidance.

THE QUEUING MANPOWER MODEL (QMAN)

INTRODUCTION

The original requirement for the Queuing Manpower Model (QMAN) arose from the search for an analytic model that would estimate maintenance manpower in the early phases of weapon system development. The Logistics Composite Model (LCOM) has been used as the maintenance manpower prediction system for the Air Force since its inception in the late 1960's (AFR 25-7, 1987). LCOM relies upon a lengthy process of constraining a complex Monte Carlo simulation to arrive at acceptable manpower estimates. This process involves running multiple simulations with varying initial manpower estimates using a mainframe computer. Estimates are changed based upon simulation outputs with optimal manpower requirements determined only after multiple runs (Boyle, 1990). Although LCOM's manpower estimates are acceptable to the Air Force's manpower community, the lengthy constraining process makes LCOM an unwieldy tool to use during the early stages of weapon system development.

Lamb, Eckstand, Seman, and Lindeman (1987) first developed an analytic approach to the manpower estimation problem with the Stochastic Process Assessment Model (SPASM). This approach relied on Stochastic Process Theory with the assumption that flightline maintenance operations were a collection of random variables that maintained a steady state. In steady state operations, the probability distribution function (PDF) of aircraft existing at a given maintenance event remains constant (Hillier & Lieberman, 1990). Lamb et al. (1987) evaluated SPASM's sortie generation and manhour estimation capabilities but did not provide evidence of the validity of the resulting manpower estimates. Further, SPASM did not take into consideration many "real-life" conditions. For example, worker slack time and utilization rate considerations were overlooked.

A more comprehensive and efficient personal computer-based model was needed. To begin this development, the actual process taking place on a present-day aircraft flightline was analyzed. QMAN was the result of this analysis. It assumes that the maintenance manpower needed will be the largest of the following values: 1) the number of personnel necessary to maintain aircraft so that the time aircraft spend awaiting maintenance and in maintenance does not prohibit them from making their next sortie; 2) the number of maintainers needed to accomplish maintenance workload in time to make the next sortie, yet prohibit over-utilization of maintainers (e.g., no maintainers may work during more than 70% of their shift); or 3) the number of personnel required so that each task, assigned to an Air Force Specialty (AFS), has enough personnel to meet its worst case crew size requirements (e.g., if there is a task that requires at least 3 people to perform then the AFS must have at least 3 personnel assigned to each shift). While the QMAN algorithm is unique, it relies heavily upon SPASM's task data compilation methods. This paper discusses the mathematical development of QMAN, the Turbo Pascal implementation of the model, the model's evaluation against LCOM, and various examples of its use.

METHODS, ASSUMPTIONS, AND PROCEDURES

Statement of the Problem

A flying organization maintains a certain number of aircraft that must fly a required number of sorties per day to meet peacetime or wartime mission requirements. These aircraft require maintenance by personnel from AFSs assigned to the flying organization's aircraft maintenance squadron. Maintenance is broken down into specific tasks, with each of these tasks assigned to a specific AFS. Specific tasks are performed in a given time, by a fixed number of personnel (crew size) assigned to the AFS. Furthermore, each task has a discrete probability of occurrence per each flying sortie. The maintainer prediction problem that QMAN addresses involves calculation of the minimum number of maintenance personnel required in each AFS to complete maintenance workload in time to meet flying demands. QMAN's approach to determining the minimum number of personnel for each AFS involves specifying the larger AFS maintainer personnel requirement from calculations based on either queuing demand theory, utilization adjusted workload computation, or maximum task crew size computation.

Queuing Theory Application

The lines that build up when aircraft are awaiting maintenance are analogous to the "bank teller problem," i.e., determining how many tellers a bank needs in order to prevent excessive customer waiting. In this problem, the distribution of the interarrival time of customers entering the bank is assumed to be exponential. This arrival distribution creates periods of high demand, with long lines that exceed teller service capability and cause high teller utilization, and periods of low demand resulting in under-utilized tellers. Queuing theory deals with problems such as these and predicts how many tellers or "servers" are necessary so that customer waiting time is not excessively long or that lines do not grow to excessive lengths.

The similarity of the aircraft maintainer prediction problem to the bank teller problem motivated the use of a queuing algorithm. In the aircraft maintainer prediction problem, aircraft replace customers and maintenance crews replace tellers. QMAN's queuing model predicts the number of crews necessary for each crew size of each AFS based upon the mean arrival rate (the average rate at which aircraft enter a crew's maintenance activity), crew service time (the average time required by a crew to perform maintenance), and the aircraft maintenance window (the time allotted for aircraft maintenance before their next sortie). Because aircraft maintenance tasks often require crews that consist of more than one individual, the queuing algorithm is applied separately to each crew size of each AFS. This is necessary in order to satisfy the constraint of the mathematical queuing model that each server be a single entity. The QMAN algorithm considers each crew as a single entityserver. Thus, AFS i has m_i different crew sizes associated with it. The QMAN algorithm is applied to each of the crew sizes, $j = 1 \dots m_i$, where crew size j consists of n_j personnel. The collection of crews of size n_j is referred to as crew size cluster j . In the model, queuing occurs within AFS i at each of the crew size clusters.

The actual number of personnel required in a specific crew size cluster is the number of crews QMAN determined to be required in the cluster times the number of personnel in a crew. The total number of personnel in a specific AFS is the sum of the number of personnel required in each of the m_i crew size clusters.

Approach

QMAN's approach to determining the minimum number of personnel for each AFS involves specifying the maintenance queuing demands, utilization adjusted workload demands, and task based crew size demands for each AFS. The actual AFS maintainer manpower requirement is the largest of these three values. Expressed mathematically, QMAN calculates A_i , the number of necessary maintenance personnel, for AFS i as:

$$A_i = \max(X_i, F_i, G_i) \quad (1)$$

where X_i = queuing demanded number of maintainers in AFS i ,
 F_i = utilization adjusted workload demanded maintainers in AFS i , and
 G_i = number of maintainers required by the task in AFS i with the largest crew size.

Number of Personnel per AFS (X_i) as a Result of Queuing Demand

The QMAN algorithm begins by calculating X_i , the number of maintainers required in each AFS to meet queuing demands. The queuing demand model calculates the number of crews for each crew size cluster in each AFS required to service the aircraft in time to meet the next scheduled sorties. Then a queuing theory model is used to calculate how much time is required for each AFS crew size cluster to perform their aircraft maintenance tasks. If this time exceeds the amount of time available for aircraft maintenance between sorties (hereafter referred to as the aircraft maintenance window), additional crews are added until the cluster's maintenance task time is less than the aircraft maintenance window. The number of personnel required in each crew size cluster is computed by multiplying the number of personnel in the crew size by the number of crews required. The total number of personnel required in an AFS is the summation of personnel required in each of the AFS's crew size clusters.

The computation of X_i involves a number of steps. First, the length of the aircraft maintenance window, denoted by the variable B , is calculated as follows. The total time available for maintenance during the day is found by subtracting the total number of hours an aircraft flies in all of its sorties (t_0 , sortie length, times R_d , the number of sorties per aircraft per day) from D , the total number of hours in the flying day. The total time available for maintenance is divided by R_d to obtain the aircraft maintenance window for each sortie.

$$B = \frac{D - (t_0 \cdot R_d)}{R_d} \quad (2)$$

Next, the total number of sorties per day is calculated by multiplying R_d , the number of sorties per aircraft per day, by α , the total number of aircraft. This value is then divided by D , the number of hours in the flying day, to yield R_h , the number of sorties per hour.

$$R_h = \frac{\alpha R_d}{D} \quad (3)$$

The total demand, C_{dij} , is the probability per aircraft sortie that crew size cluster j of AFS i will need to perform maintenance and is calculated by adding all of the probabilities of occurrence, p_k , for the maintenance tasks performed by that particular AFS/crew size cluster (Lamb et al., 1987).

$$C_{dij} = \sum_{k=1}^{y_{ij}} p_k, \text{ for crew size cluster } j \text{ of AFS } i \quad (4)$$

where y_{ij} = the total number of maintenance tasks for crew size cluster j of AFS i

Similarly, the total demand weighted service time, C_{tij} , that is, the average maintenance time per aircraft sortie for crew size cluster j of AFS i , is calculated by finding the total work time required for cluster j and dividing this value by the total demand for the AFS/crew size cluster, C_{dij} . The total work time for the cluster is found by summing the products of the probabilities, p_k , and the maintenance times, t_k , for the y_{ij} different tasks associated with crew size cluster j of AFS i .

$$C_{tij} = \frac{\sum_{k=1}^{y_{ij}} p_k t_k}{C_{dij}}, \text{ for crew cluster } j \text{ of AFS } i \quad (5)$$

Next, M_i , the required man-hours for AFS i is calculated by summing the products of the crew sizes, n_j , the task probabilities, p_k , and the task times, t_k , for the y_{ij} different tasks for each of the m_i different crew cluster sizes associated with AFS i (Lamb et al., 1987).

$$M_i = \sum_{j=1}^{m_i} \sum_{k=1}^{y_{ij}} n_j p_{kj} t_{kj}, \text{ for each AFS } i. \quad (6)$$

Using the above calculations a series of other queuing variables that are required can now be calculated. The first of these is the mean arrival rate, λ_{ij} , that is, the average rate at which aircraft arrive for maintenance to crew size cluster j of AFS i . To calculate λ_{ij} , the cumulative demand, C_{dij} , is multiplied by the number of sorties per hour, R_h .

$$\lambda_{ij} = R_h C_{dij} \quad (7)$$

The mean service rate, μ_{ij} , i.e., the expected number of aircraft completing service per hour, is the inverse of C_{tij} , the total demand weighted service time for crew size cluster j of AFS i .

$$\mu_{ij} = \frac{1}{C_{ij}} \quad (8)$$

The total number of crews, s_{ij} , required for crew cluster j in AFS i is initialized to the smallest feasible number of crews that can produce steady-state operation. This quantity represents the number of personnel necessary to maintain a constant probability density function (PDF) of aircraft at each AFS/crew size cluster maintenance event.

$$s_{ij} = \text{Truncate}\left(\frac{\lambda_{ij}}{\mu_{ij}}\right) + 1 \quad (9)$$

With the mean arrival rate, λ_{ij} , the mean service rate, μ_{ij} , and the total number of crews, s_{ij} , now determined, the expected maintenance time, W_{ij} , for an aircraft requiring maintenance from crew size j of AFS i can be calculated. If s_{ij} is equal to one, then

$$W_{ij} = \frac{1}{\mu_{ij} - \lambda_{ij}} \quad (10)$$

However, when more than one crew is required, the calculation of W_{ij} becomes more complicated. The utilization factor for the servers (ρ), the probability that no customers are in the queuing system (P_0), the expected queue length (L_q), and the expected waiting time in the queue (W_q) are introduced and determined as follows (Hillier & Lieberman, 1990):

$$\rho_{ij} = \frac{\lambda_{ij}}{\mu_{ij}s_{ij}} \quad (11)$$

$$P_{0ij} = \frac{1}{\left[\sum_{\alpha=0}^{s_{ij}-1} \frac{(\lambda_{ij} / \mu_{ij})^\alpha}{\alpha!} + \left(\frac{(\lambda_{ij} / \mu_{ij})^{s_{ij}}}{s_{ij}!} \cdot \frac{1}{1 - (\lambda_{ij} / s_{ij}\mu_{ij})} \right) \right]} \quad (12)$$

$$L_{qij} = \frac{P_{0ij}(\lambda_{ij} / \mu_{ij})^{s_{ij}} \rho_{ij}}{s_{ij}!(1 - \rho_{ij})^2} \quad (13)$$

$$W_{qij} = \frac{L_{qij}}{\lambda_{ij}} \quad (14)$$

where α = number of aircraft, and
 s_{ij} = number of crews in crew size cluster j of AFS i

Then the expected maintenance time, W_{ij} , which includes the time an aircraft spends waiting for maintenance, is calculated by:

$$W_{ij} = W_{qij} + \frac{1}{\mu_{ij}} \quad (15)$$

To determine s_{ij} , W_{ij} , the expected maintenance time is compared to B , the aircraft maintenance window. If the W_{ij} is less than B , then all maintenance actions required of crew size cluster j of AFS i can be completed within the window and s_{ij} is not adjusted. If the W_{ij} is greater than B , more crews of crew size cluster j are needed to accomplish the required maintenance. In this case, s_{ij} is increased by one and W_{ij} is recalculated and compared to B . This process of incrementally increasing s_{ij} by one is repeated until W_{ij} is less than B .

The number of personnel in crew size cluster j of AFS i , denoted as X_{ij} , required to meet queuing demand is determined by multiplying s_{ij} , the number of required crews in crew size cluster j of AFS i , by n_j , the size of crew size cluster j .

$$X_{ij} = s_{ij} \cdot n_j \quad (16)$$

The total number of maintainers needed in AFS i as a result of queuing demand can then be determined by summing the number of personnel required in each of the m_i crew size clusters of AFS i .

$$X_i = \sum_{j=1}^{m_i} X_{ij} \quad (17)$$

Number of Personnel per AFS (F_i) as a Result of Utilization Adjusted Workload Effects

The present QMAN queuing demand computations do not account for reduced personnel availability due to non-maintenance workload such as personnel supervision, training and other administrative functions. Nor do they account for worker slack time due to part and aircraft non-availability. Thus actual worker utilization rates, i.e., the percentage of a worker's time actually spent working on aircraft compared to the workers total time available to do work, are below those predicted by QMAN. Since very few AFSs have actual utilization rates above 70%, QMAN allows the user to set a limit on the utilization rates by assigning a maximum utilization rate, U , for all AFSs. The minimum number of personnel in AFS i required to conduct the workload necessary to maintain a desired sortie rate at or below the maximum utilization rate, U , is calculated by:

$$F_i = \frac{100(M_i R_h)}{U} \quad (18)$$

where U is as a percentage between 0 and 100.

Number of Personnel per AFS (G_i) as a Result of Required Maximum Crew Sizes

It is essential that each AFS has enough personnel to accomplish all assigned tasks. QMAN's maximum AFS crew size assessment establishes a minimum value for the number of personnel in an AFS. This value, G_i , is the number of personnel in the largest crew size.

Assumptions of Model

In the development of QMAN it was necessary to make three assumptions that do not completely to the reality of aircraft maintenance. These assumptions are:

1. The interarrival times for aircraft arriving to each crew size cluster of each AFS are exponentially distributed and these distributions are constant with time. This steady state assumption oversimplifies the aircraft maintainer problem because actual sorties are flown in various complex "batch" patterns which produce exponential maintenance arrival distributions that are not constant through time. However, for the purposes of manpower requirements estimation this assumption is acceptable and, as will be shown, produces results very close to those produced by LCOM, a discrete-event simulation of actual aircraft sortie generation.

2. The maintenance task times are exponentially distributed and all maintenance tasks are independent of each other. In reality, maintenance task times tend to be distributed lognormally. Two facts, however, make this distribution substitution acceptable: simulations of aircraft maintenance activities normally contain large numbers of each maintenance action and QMAN only makes use of each maintenance action's average task time. The Central Limit Theorem states that, given a large enough sample of a random variable, the distribution of the samples around the mean is normal, regardless of the underlying distribution of the random variable (Hogg, 1980).

3. Maintenance on an aircraft may be performed simultaneously (in parallel) by members (crew size clusters) of a single AFS or of different AFSs. In actual day-to-day operations some maintenance actions cannot be performed simultaneously. For example, due to safety considerations, maintenance on an aircraft's fuel systems precludes the performance of all other maintenance activities. However, these cases are the exception, not the rule; the majority of aircraft maintenance actions can be performed simultaneously.

IMPLEMENTATION

QMAN was implemented in Borland's Turbo Pascal for Microsoft Windows (Appendix A contains QMAN source code) following the development of the theoretical model. QMAN will run on any 80386 or 80486-based computer equipped with Microsoft Windows 3.0 or higher. Input files must be developed by the user using LCOM "hit matrices" that contain individual records for each task assigned to an AFS. Each record contains the AFS to which the task is assigned, the mean task time, the crew size required to complete the task, and the task's

probability of occurrence per sortie (Fig 1). The field labeled "Task Type/Priority" gives information concerning whether the task is performed on or off the aircraft.

AFS	TASK	Mean Time To Perform Task	Crew Size	Probability of Task Occurrence	Task Type/ Priority
431R1	T13H00	1.00	2	0.002266	21
431R1	T14G00	2.00	2	0.000302	21
461S0	R75H6B	2.37	2	0.000151	21
462L0	BCH1	0.10	3	0.379211	31
462L0	BFL1	0.10	3	0.085814	31
462L0	BGUN	0.10	3	0.156217	31
462L0	BSET1	0.30	3	0.478169	31
462L0	ICT	0.30	3	0.373017	31
462X0	R11D04	1.20	2	0.002266	21
462X0	M11L03	3.50	2	0.000151	21

Figure 1. QMAN Input File

After these values are read, QMAN prompts the user for information on the number of aircraft, the aircraft sortie rate, sortie length, and maximum allowable utilization rates. QMAN then computes X_i , the number of personnel required for each AFS due to queuing, and F_i , the number required to meet utilization requirements AFSs that have manpower determined by the maximum crew size in the AFS rather than queuing or utilization demands (i.e., $G_i > F_i$ and $G_i > X_i$), are flagged "driven by crew." This enables the user to target AFSs that may be candidates for specialty restructuring. Combining AFSs with low utilization could reduce overall manpower requirements by creating a single AFS with higher utilization but fewer people. Output information is either written to the screen (Fig 2) or to a file for further analysis.

326X8	10.619	
404S1	2.000	**DRIVEN BY CREW**
423S2	2.000	**DRIVEN BY CREW**
423S3	6.427	
423X0	5.956	
423X1	2.024	
423X4	6.000	
426X2	38.000	
427S5	2.000	**DRIVEN BY CREW**
427X0	2.000	**DRIVEN BY CREW**
427X5	7.832	
431E1	33.020	
431F1	5.959	
431R1	6.029	
461D0	10.022	
461M0	32.054	
461S0	12.381	
462E0	4.000	
462L0	32.029	
462X0	13.062	
4BRIF	3.860	
TOTAL		259.962
Press ENTER to return to the menu...		

Figure 2. QMAN Output Screen

EVALUATION

QMAN was validated by comparing its manpower estimates with those derived from LCOM simulations of steady state flightline scenarios. A traditional validation design involving a direct comparison of manpower estimates independently derived from the two models was not feasible due to LCOM run-time constraints. Instead, QMAN manpower estimates were obtained for the 70% maximum utilization rate and were then used as a starting point for the LCOM constraining process.¹ The objective was to determine if, after appropriate constraining, the sortie rate achieved by LCOM was comparable to that for the initial QMAN estimate.

The evaluation procedures were accomplished for numerous scenarios involving different sortie rates and sortie lengths for three aircraft types. The Advanced Tactical Fighter (now designated as F-22), the F-15, and the F-16D served as the aircraft modeled in the evaluations. Additionally, QMAN runs were made at varying utilization rates to determine the utilization level that produced estimates most similar to those achieved by LCOM. All QMAN input files were consistent with LCOM input with respect to task times, task probabilities, and crew size information.

The results were favorable across the various scenarios; QMAN estimates were consistently comparable to those obtained from LCOM. To illustrate the first evaluation, the results for one scenario are presented in Table 1 and are described in detail here. The table shows the manpower estimates from QMAN runs employing different maximum utilization rates (60%, 70%, 80%, 90%, and 100%), as well as the LCOM estimate required to achieve the QMAN sortie rate. The scenario examined was for 24 F-22 aircraft flying three 2-hour sorties per aircraft per day. The total number of manpower positions required, and the mean and standard deviation of the estimates, for 17 AFSs are shown.

Two major findings are noteworthy. First, QMAN results showed the expected decrease in the number of positions required as the manpower utilization rate improved. That is, there was a consistent decrease in manpower requirements from 64 to 53 positions as the utilization rate increased from 60% to 100%. Second, throughout the utilization range, QMAN produced estimates that were similar to those produced by LCOM. Further, the smallest difference in manpower estimates was observed at the 70% maximum utilization rate used to initiate the LCOM simulations. The QMAN estimate of 59 positions total (or an average of 3.47 positions in 17 AFSs) was extremely close to the LCOM estimate of the manpower requirements to achieve the sortie rate (60 positions total or an average of 3.53 positions across AFSs). Thus, only very minor changes to the QMAN manpower estimates were necessary to achieve similar sortie rates in LCOM.

The results in Table 1 are representative of those for the remainder of the conditions examined. As shown by the supporting data in Appendix B for the F-22 aircraft and in Appendix

¹ Since LCOM is a Monte Carlo simulation it does not produce specific manpower estimates as a result of a single run, rather, various heuristics and numerous runs are needed to constrain the model to targeted sortie rates producing acceptable manpower estimates.

C for the F-15 aircraft, the high level of accuracy achieved by QMAN was stable for different scenarios.

Table 1. Comparison of QMAN and LCOM Manpower Estimates
24 F-22 aircraft flying three 2-hour sorties per aircraft per day (N=17)

	QMAN-60%	QMAN-70%	QMAN-80%	QMAN-90%	QMAN-100%	
AFSC	Max Utilization	Max Utilization	Max Utilization	Max Utilization	Max Utilization	LCOM
326X6	9	8	7	6	6	8
327X7	3	2	2	2	2	2
328X8	2	2	2	2	2	2
404X1	2	2	2	2	2	2
423X0	3	3	3	3	3	3
423X1	3	3	3	3	3	3
423X2	2	2	2	2	2	2
423X3	5	5	4	4	4	5
423X4	3	3	3	3	3	3
426X2	6	5	4	4	4	5
427X1	1	1	1	1	1	1
427X2	2	2	2	2	2	2
427X5	4	3	3	3	3	4
431F1 ^a	8	7	6	5	5	7
431R1 ^a	2	2	2	2	2	2
462LO ^a	6	6	6	6	6	6
462X0 ^a	3	3	3	3	3	3
Total	64	59	55	53	53	60
Mean	3.76	3.47	3.24	3.12	3.12	3.53
SD	2.28	2.00	1.68	1.45	1.45	2.00

^a A letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

EXAMPLES OF QMAN APPLICATIONS

The ability of QMAN to approximate LCOM results makes possible various types of manpower analyses that previously would have been extremely time consuming. Two examples of the types of manpower impact studies possible using QMAN are described below.

Effect of an Increase in Maintainer Productivity on Manpower

In order to determine the effect of an increase in maintainer productivity on manpower requirements, several QMAN runs were made with varying constants multiplied by task performance times. These constants allowed an increase in maintainer productivity to be represented by a decrease in task performance time.

This illustrative scenario used the F-15 again flying three 2-hour sorties per aircraft per day. Other model parameters were 72 aircraft, a 24-hour flying day, and a maximum maintainer

utilization of 70%. This example employed three AFSs: 326X8 (Avionics and Communications Technician), 423X3 (Fuel Systems Technician), and 472X0 (Special Vehicle Helper). The behavior of these AFSs was representative of the remainder of the AFSs.

As maintainer task performance times decreased, the number of maintainers necessary to support flying demands decreased (Table 2). These decreases in number of required maintainers were not uniform for all AFSs. Certain AFSs showed greater manpower savings than others.

Table 2. Effect of Increase in Productivity on Number of Manpower Positions Required

AFS	0% (Baseline)	Percentage Increase in Maintainer Productivity			
		+10%	+20%	+30%	+40%
326X8	11	10	9	8	7
423S3	7	6	6	5	5
472X0	2	2	2	2	2

For example, 326X8 saved 4 positions with a 40% increase in maintainer productivity while 472X0 did not see any savings with a similar increase in productivity. This effect occurred because crew size requirements, G_i , determined manpower for 472X0 (i.e., $G_i > F_i$, $G_i > X_i$). No matter how fast maintainers work, AFS 472X0 must have at least two people in order to accomplish assigned tasks.

This type of analysis capability makes it possible to quantify the manpower savings associated with increases in maintainer productivity. Additionally, it provides information as to which AFSs show the greatest potential savings as a result of increased maintainer productivity.

Effect of Aircraft Reliability on Aircraft Maintainer Requirements

The second illustrative example shows the effect of changing aircraft reliability on manpower requirements. This manpower savings is computable by aircraft component to show the impact of increased equipment reliability on manpower requirements. Weapon designers could use this information to determine which pieces of equipment will yield the greatest decrease in manpower as a result of increased reliability. Designers could then engineer these components for increased reliability to maximize the use of limited weapon acquisition resources.

Using the same F-15 scenario as in the previous example, reliability was decreased and increased by varying the probability of part failure per sortie for all tasks associated with a certain component. This process was repeated for all components. While manpower requirements decreased with increased reliability over all components, increased reliability of certain aircraft systems yielded greater savings than others. Radio Navigation with a work unit code (71), Radar/Fire System with a work unit code of (74), and Weapons Delivery System with a work unit code of (75) each experienced decreased manpower requirements with increased reliability

(Figure 3). The largest savings was seen in the Weapons Delivery System. Therefore, according to the data presented in this example, weapon designers could achieve the greatest potential manpower savings from investing in development processes to increase the reliability of the Weapons Delivery System.

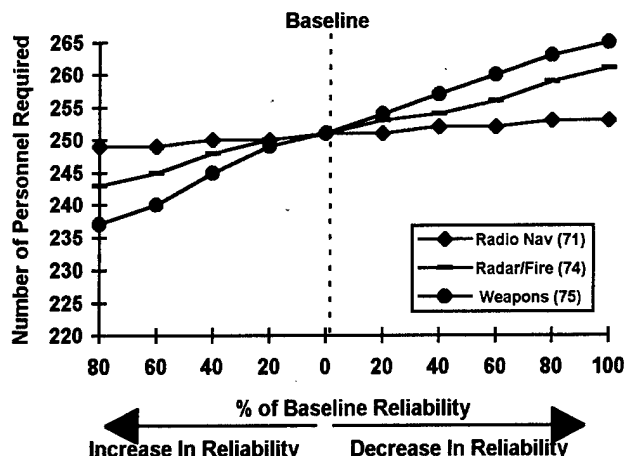


Figure 3. Effect of Aircraft Reliability on Manpower Requirements

CONCLUSION

The Queuing Manpower Model (QMAN) is an effective manpower estimator with results comparable to LCOM for steady state operational unit flying conditions. When QMAN manpower estimates are used in LCOM simulation, sortie rates very close to those specified in the QMAN model are achieved. Since run-times are normally less than one second for this analytic model, QMAN can be applied efficiently to a variety of manpower analysis problems that, due its lengthy simulation run-times, are overly time consuming if analyzed using LCOM. Examples include determining the effects of an increase in maintainer productivity and changes in aircraft reliability on aircraft maintainer requirements.

Despite its strengths, QMAN is not intended as a replacement for LCOM but rather as a complement to it. QMAN does not consider many of the uncertainties of actual flightline maintenance. These complexities can only be captured using discrete-event simulation methods. Furthermore, QMAN can only determine manpower requirements for simple, steady state flying conditions. This steady state assumption oversimplifies the aircraft maintainer problem because actual sorties are flown in various complex "batch" patterns. This type of sortie generation produces exponential maintenance arrival distributions that are **not** constant through time. While the capability to model non-steady state flightline conditions is not necessary for the trade-off analyses QMAN performs, it is essential for developing actual manpower requirements the Air Force uses to set manning standards.

Finally, use of QMAN to assess and reduce manpower requirements in the early stages of weapon system acquisition or modification could be a major component in the Air Force's program to design and acquire weapons at the lowest life cycle cost.

REFERENCES

- Boyle, E.B. (1990). Logistics Composite Model Explained (AFHRL-TP-90-58). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.
- Department of the Air Force. Logistics Composite Modeling. AFR 25-7. Washington: HQ USAF, 1987.
- Hillier, F.S. & Lieberman, G.J. (1990). Introduction to Operations Research (5th ed.). New York: McGraw-Hill Publishing Company.
- Hogg, R.T. & Craig, A.T. (1986). Introduction to Mathematical Statistics (4th ed.). New York: Macmillan Publishing Company, Inc.
- Lamb, T.A., Eckstand, G.A., Seman, T.R., & Lindeman, R.A. (1987). Small Unit Maintenance Specialties for the F-16: Task Identification, Data Base Development, and Exploratory Cluster Analyses (AFHRL-TP-87-23, AD-B118 623L). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

APPENDIX A: QMAN SOURCE CODE

```
(*****)  
(* PROGRAM : QMAN  
(* DATE   : September 15, 1992  
(* AUTHORS : 1LT Jeff Grobman, 2LT David Quick, and Cadet Russ Workman  
(*  
(* PURPOSE : To use an analytic queueing model to determine flightline  
(*           manpower requirements.  
(*  
(* PROCEDURES : GETINFO, ACCUMULATE, CALCULATE, GETPEOPLE, GETPOS,  
(*             UTILIZATION, BOOST, REPORT, HIDRIVER, INDIVIDUAL  
(*****)  
PROGRAM QMAN;  
USES  
    WINCRT;  
  
CONST  
    MAXAFSCLENGTH=50;  
    MAXCREW=5;  
    MAXEQUIPLGTH = 200;  
  
TYPE  
    ARRAYTYPE = ARRAY[1..MAXAFSCLENGTH] OF DOUBLE;  
    TASKTYPE = ARRAY[1..MAXAFSCLENGTH] OF STRING[5];  
    CLUSTYPE = ARRAY[1..MAXAFSCLENGTH,1..MAXCREW] OF DOUBLE;  
    BOOLTTYPE = ARRAY[1..MAXAFSCLENGTH] OF BOOLEAN;  
    EQUIPTYPE = ARRAY[1..MAXEQUIPLGTH] OF STRING[2];  
    EQTYPE = STRING[2];  
  
MAN=RECORD  
    AFS : TASKTYPE;  
    POWER : ARRAYTYPE;  
    MINCREW : ARRAYTYPE;  
    MANH : ARRAYTYPE;  
    UT : ARRAYTYPE;  
    FLAG : BOOLTTYPE;  
END;  
  
VAR  
    ALPHA : DOUBLE;  
    URATE : DOUBLE;  
    COUNT : INTEGER;  
    COUNT2 : INTEGER;  
    WIDTH : INTEGER;
```

```

AIRMAN : MAN;
CTIME : CLUSTYPE;
CLUSTIME : CLUSTYPE;
CLUSDEMAND : CLUSTYPE;
SORTIERATE: DOUBLE;
WINDOW : DOUBLE;
NEWSET : STRING;
ANS : CHAR;
POWER : ARRAYTYPE;
TOTPOW : DOUBLE;
E : EQUIPTYPE;
EQ : EQTYPE;
SELECT : INTEGER;
READY1 : BOOLEAN;
READY2 : BOOLEAN;

```

```

(*****
PROCEDURE INTRO;

```

```

BEGIN (*INTRO*)
  CURSORTO(0,5);
  WRITELN('*****');
  WRITELN('*');
  WRITELN('*');
  WRITELN('*      QMAN      *');
  WRITELN('*    version 1.0    *');
  WRITELN('*');
  WRITELN('*');
  WRITELN('*****');
  WRITELN;
  WRITELN;
  WRITELN;
  WRITE('Press ENTER to continue...');
  READLN;
END; (*INTRO*)

```

```

(*****
PROCEDURE MENU(VAR SELECT : INTEGER);

```

```

BEGIN (*MENU*)
  CLRSCR;
  CURSORTO(0,5);
  WRITELN('  MENU');
  WRITELN('  *****');
  WRITELN('(1) Choose weapon system');

```

```

WRITELN('(2) Choose initial parameters');
WRITELN('(3) Manpower Report');
WRITELN('(4) Utilization Report');
WRITELN('(5) Hi Driver Report');
WRITELN('(6) Reliability by AFSC');
WRITELN('(7) Reliability by Equipage');
WRITELN('(8) Combine AFSC"s');
WRITELN('(9) Exit');
WRITELN;
WRITELN;
WRITELN('*** (1) and (2) have to be the first entries!***');
WRITELN;
WRITE('Make selection and press ENTER: ');
READLN(SELECT);
END; (*MENU*)

```

```

(*****)
PROCEDURE WEAPON(VAR NEWSET : STRING;
VAR READY1 : BOOLEAN);

```

```

VAR
WS : INTEGER;

```

```

BEGIN (*WEAPON*)
CLRSCR;
CURSORTO(0,5);
WRITELN(' Weapon Systems');
WRITELN(' *****');
WRITELN('(1) ATF');
WRITELN('(2) F-15');
WRITELN;
WRITELN;
WRITE('Make selection and press ENTER: ');
READLN(WS);
CASE WS OF
1 : NEWSET := 'C:\QMAN\ATFON.DAT';
2 : NEWSET := 'C:\QMAN\ON2.DAT';
END;
READY1 := TRUE;
END; (*WEAPON*)

```

```

(*****)
(* PROCEDURE : GETINFO
(*
(* PURPOSE : Allows the user to pick his data set and initial
(* parameters. Performs basic calculations to set a

```

```

(*      window and to adjust the sortierate.
(*
(* INPUT VARIABLES : None
(*
(* OUTPUT VARIABLES :
(*  NEWSET - path of the data set to be used
(*  SORTIERATE - number of sorties per hour
(*  WINDOW - a calculated time in which maintainance
(*    has to be done
(*  ALPHA - total number of aircraft
(*  URATE - maximum utilization for any AFSC
(*
(* LOCAL VARIABLES :
(*  DAY - number of flying hours in the day
(*  SORTIELENGTH - length of each sortie
(*  CHANGE - boolean that represents a change or no change
(*    to the data set
(*****
PROCEDURE GETINFO(VAR SORTIERATE:DOUBLE;
                  VAR WINDOW:DOUBLE;
                  VAR ALPHA : DOUBLE;
                  VAR URATE : DOUBLE;
                  VAR READY2 : BOOLEAN);

VAR
  DAY : DOUBLE;
  SORTIELENGTH:DOUBLE;

BEGIN (*GETINFO*)
  CLRSCR;
  CURSORTO(0,5);
  WRITELN('  Initial Parameters');
  WRITELN(' *****');
  WRITE('Enter the total number of aircraft: ');
  READLN(ALPHA);
  WRITE('Enter the sortie rate per aircraft per day: ');
  READLN(SORTIERATE);
  WRITE('Enter the sortie length: ');
  READLN(SORTIELENGTH);
  WRITE('Enter the total flying day: ');
  READLN(DAY);
  WRITE('Enter the maximum utilization rate: ');
  READLN(URATE);
  WINDOW:=((DAY-(SORTIERATE*SORTIELENGTH))/SORTIERATE);
  SORTIERATE:=((ALPHA/DAY)*SORTIERATE);
  READY2 := TRUE;

```

END; (*GETINFO*)

(*****)

(* PROCEDURE : ACCUMULATE

(*

(* PURPOSE : Reads the input file and determines if solution
(* possible.

(*

(* INPUT VARIABLES : None

(*

(* OUTPUT VARIABLES :

(* AIRMAN - a record data specific for each AFSC that includes:

(* AFS - AFSC

(* POWER - required manpower

(* MINCREW - smallest crewsize

(* MANH - required manhours

(* UT - utilization

(* FLAG - boolean which stops iteration when additional men
(* have little effect on time

(* CLUSTIME - time needed for specific AFSC and crewsize

(* CLUSDEMAND - demand for specific AFSC and crewsize

(* COUNT - total number of AFSCs

(* WIDTH - crewsize

(*

(* LOCAL VARIABLES :

(* INPUT1 - input file

(* AFSC, TASK, TIME, LAMBDA, CREW - information being read
(* from the input file

(* OLD - temporarily holds the current AFSC

(* BLANK1, BLANK2 - hold spaces so the file can be read

(* I, J - counters

(* TOTAL - total maintainance time required

(*****)

PROCEDURE ACCUMULATE(VAR AIRMAN : MAN;

VAR CTIME : CLUSTYPE;

VAR CLUSTIME : CLUSTYPE;

VAR CLUSDEMAND : CLUSTYPE;

VAR COUNT : INTEGER;

VAR WIDTH : INTEGER);

VAR

INPUT1 : TEXT;

AFSC : STRING[6];

OLD : STRING[6];

TASK : STRING[8];


```

TIME : DOUBLE;
CREW : INTEGER;
LAMBDA : DOUBLE;
BLANK1 : STRING[1];
BLANK2 : STRING[1];
I : INTEGER;
J : INTEGER;
alley:real;

```

```

BEGIN (*ACCUMULATE*)
alley:=1;(******for sortierates******)
  FOR I:=1 TO MAXAFSCLENGTH DO
    BEGIN (*FOR1*)
      AIRMAN.MINCREW[I] := 0;
      AIRMAN.MANH[I] := 0;
      FOR J:= 1 TO MAXCREW DO
        BEGIN (*FOR2*)
          CTIME[I,J]:=0;
          CLUSTIME[I,J]:=0;
          CLUSDEMAND[I,J]:=0;
        END; (*FOR2*)
      END; (*FOR1*)
      COUNT := 0;
      WIDTH := 0;
      ASSIGN(INPUT1, NEWSET);
      RESET(INPUT1);
      WHILE NOT EOF(INPUT1) DO
        BEGIN (*WHILE*)
          OLD:=AFSC;
          READLN(INPUT1,AFSC,BLANK1,TASK,BLANK2,TIME,CREW,LAMBDA);
          time:=time*alley;
          IF OLD<>AFSC THEN
            COUNT:=COUNT+1;
            AIRMAN.AFS[COUNT]:=AFSC;
            IF AIRMAN.MINCREW[COUNT]<CREW THEN
              AIRMAN.MINCREW[COUNT]:=CREW;
            CTIME[COUNT,CREW]:=CTIME[COUNT,CREW]+(LAMBDA*TIME);
            CLUSDEMAND[COUNT,CREW]:=CLUSDEMAND[COUNT,CREW]+LAMBDA;
            IF CREW > WIDTH THEN
              WIDTH:=CREW;
            AIRMAN.MANH[COUNT]:=AIRMAN.MANH[COUNT]+(CREW*TIME*LAMBDA);
          END; (*WHILE*)
        FOR I:= 1 TO MAXAFSCLENGTH DO
          FOR J:= 1 TO MAXCREW DO
            IF CLUSDEMAND[I,J]>0 THEN

```

```

        CLUSTIME[I,J]:=CTIME[I,J]/CLUSDEMAND[I,J];
    CLOSE(INPUT1);
END; (*ACCUMULATE*)

```

```

    (*****
    (* PROCEDURE : CALCULATE
    (*
    (* PURPOSE : Calculates the expected waiting time in the system,
    (*   or length of the window.
    (*
    (* INPUT VARIABLES :
    (*   S - number of crews, or servers
    (*   LAMBDA - mean arrival rate of planes
    (*   U - mean service rate
    (*
    (* OUTPUT VARIABLES :
    (*   W - expected waiting time in system
    (*
    (* LOCAL VARIABLES :
    (*   I - counter
    (*   P - utilization factor for the servers
    (*   TEMP1, TEMP2, D1, D2, D3 - temporary values used for
    (*   calculations
    (*   P0 - probability that no planes are in the queueing system
    (*   LQ - expected queue length(excludes planes being serviced)
    (*   WQ - expected waiting time in queue
    (*****

```

```

PROCEDURE CALCULATE(VAR W : DOUBLE;
                    S : DOUBLE;
                    LAMBDA : DOUBLE;
                    U : DOUBLE);

```

```

VAR
    I : INTEGER;
    P : DOUBLE;
    TEMP1 : DOUBLE;
    TEMP2 : DOUBLE;
    D1 : DOUBLE;
    D2 : DOUBLE;
    D3 : DOUBLE;
    P0 : DOUBLE;
    LQ : DOUBLE;
    WQ : DOUBLE;

```

```

BEGIN (*CALCULATE*)

```

```

IF S = 1 THEN
  W := 1/(U-LAMBDA)
ELSE
  BEGIN (*ELSE*)
    P := LAMBDA/(U*S);
    IF P = 1 THEN P := LAMBDA/(U*(S+0.1));
    TEMP1 := 1;
    TEMP2 := 1;
    D1 := 1;
    FOR I := 1 TO TRUNC(S-1) DO
      BEGIN (*FOR1*)
        TEMP1 := (LAMBDA/U)*TEMP1;
        TEMP2 := I*TEMP2;
        D1 := D1 + TEMP1/TEMP2;
      END; (*FOR1*)
    TEMP1 := 1;
    TEMP2 := 1;
    FOR I := 1 TO TRUNC(S) DO
      BEGIN (*FOR2*)
        TEMP1 := (LAMBDA/U)*TEMP1;
        TEMP2 := I*TEMP2;
      END; (*FOR2*)
    D2 := TEMP1/TEMP2;
    D3 := 1/(1-P);
    P0 := 1/(D1 + D2*D3);
    LQ := (P0*TEMP1*P)/(TEMP2*(1-P)*(1-P));
    WQ := LQ/LAMBDA;
    W := WQ + 1/U;
  END; (*ELSE*)
END; (*CALCULATE*)

```

```

(*****)
(* PROCEDURE : GETPEOPLE
(*
(* PURPOSE : Adds men until the downtime is less than the window.
(*
(* INPUT VARIABLES :
(*   LAMBDA - mean arrival rate of planes
(*   TIME - service time for an AFSC cluster
(*
(* OUTPUT VARIABLES :
(*   PEOPLE - number of crews needed for an AFSC cluster
(*   TEMPFLAG - warning that additional men will have little or
(*     no effect on downtime
(*

```

```

(* LOCAL VARIABLES :
(*  DONE - boolean that stops iteration
(*  COUNT - keeps track of number of iterations
(*  U - mean service rate
(*  MEN - number of crews, or servers
(*  DOWNTIME - expected waiting time in system
(*  MINMEN - starting value for men
(*  TEMPDOWN - downtime for (MEN - 1)
(*****)

```

```

PROCEDURE GETPEOPLE(VAR PEOPLE : DOUBLE;
    VAR TEMPFLAG : BOOLEAN;
    LAMBDA : DOUBLE;
    TIME : DOUBLE);

```

```

VAR
    DONE : BOOLEAN;
    COUNT : INTEGER;
    U : DOUBLE;
    MEN : DOUBLE;
    DOWNTIME : DOUBLE;
    MINMEN : DOUBLE;
    TEMPDOWN : DOUBLE;

```

```

BEGIN (*GETPEOPLE*)
    DONE := FALSE;
    U := 1/TIME;
    TEMPDOWN := 0;
    COUNT := 0;
    MINMEN := LAMBDA/U;
    MEN := TRUNC(MINMEN) + 1;
    WHILE DONE = FALSE DO
        BEGIN (*WHILE*)
            CALCULATE(DOWNTIME,MEN,LAMBDA,U);
            IF ((TEMPDOWN-DOWNTIME) < 0.017) AND (COUNT > 0) THEN
                BEGIN (*IF1*)
                    DONE := TRUE;
                    TEMPFLAG := TRUE;
                END; (*IF1*)
            IF DOWNTIME > WINDOW THEN
                BEGIN (*IF2*)
                    TEMPDOWN := DOWNTIME;
                    MEN := MEN + 1;
                    COUNT := COUNT + 1;
                END (*IF2*)
            ELSE

```

```

        BEGIN (*ELSE*)
            DONE := TRUE;
            IF MEN = 1 THEN
                MEN := MINMEN;
            END; (*ELSE*)
        END; (*WHILE*)
        PEOPLE := MEN;
    END; (*GETPEOPLE*)

    (*****
    (* PROCEDURE : GETPOS
    (*
    (* PURPOSE : Computes manpower for all of the AFSCs.
    (*
    (* INPUT VARIABLES : None
    (*
    (* OUTPUT VARIABLES :
    (*   AIRMAN - a record data specific for each AFSC that includes:
    (*       AFS - AFSC
    (*       POWER - required manpower
    (*       MINCREW - smallest crewsize
    (*       MANH - required manhours
    (*       UT - utilization
    (*       FLAG - boolean which stops iteration when additional men
    (*           have little effect on time
    (*
    (* LOCAL VARIABLES :
    (*   I, J - counters
    (*   TEMP - mean arrival rate of planes
    (*   PEOPLE - number of crews needed for an AFSC cluster
    (*   TEMPFLAG - warning that additional men will have little or
    (*       no effect on downtime
    (*****
    PROCEDURE GETPOS(VAR AIRMAN : MAN);

    VAR
        I : INTEGER;
        J : INTEGER;
        TEMP : DOUBLE;
        PEOPLE : DOUBLE;
        TEMPFLAG : BOOLEAN;

    BEGIN (*GETPOS*)
        FOR I:= 1 TO COUNT DO
            BEGIN (*FOR1*)

```

```

    PEOPLE := 0;
    AIRMAN.POWER[I]:=0;
    AIRMAN.FLAG[I] := FALSE;
    FOR J:= 1 TO WIDTH DO
        BEGIN (*FOR2*)
            TEMPFLAG := FALSE;
            TEMP := SORTIERATE*CLUSDEMAND[I,J];
            IF ((CLUSTIME[I,J]<>0)AND(CLUDEMAND[I,J]>0.00001))THEN
                BEGIN (*IF*)
                    GETPEOPLE(PEOPLE,TEMPFLAG,TEMP,CLUSTIME[I,J]);
                    PEOPLE := PEOPLE*J;
                    IF TEMPFLAG = TRUE THEN
                        AIRMAN.FLAG[I] := TEMPFLAG;
                        AIRMAN.POWER[I]:=AIRMAN.POWER[I]+PEOPLE;
                    END; (*IF*)
                END; (*FOR2*)
            END; (*FOR1*)
        END; (*GETPOS*)

    (*****
    (* PROCEDURE : BOOST
    (*
    (* PURPOSE : Increases men for each AFSC until the utilization
    (*   gets below the user defined maximum utilization.
    (*
    (* INPUT VARIABLES : None
    (*
    (* OUTPUT VARIABLES :
    (*   AIRMAN - a record data specific for each AFSC that includes:
    (*     AFS - AFSC
    (*     POWER - required manpower
    (*     MINCREW - smallest crewsize
    (*     MANH - required manhours
    (*     UT - utilization
    (*     FLAG - boolean which stops iteration when additional men
    (*       have little effect on time
    (*
    (* LOCAL VARIABLES :
    (*   I - counter
    (*   NEWMAN - number of men needed not considering queueing
    (*     effects
    (*****
    PROCEDURE BOOST(VAR POWER : ARRAYTYPE;
                    VAR TOTPOW : DOUBLE);

```

```

VAR
  I : INTEGER;
  NEWMAN : DOUBLE;

BEGIN (*BOOST*)
  NEWMAN := 0;
  TOTPOW := 0;
  FOR I:= 1 TO COUNT DO
    BEGIN (*FOR*)
      NEWMAN := AIRMAN.MANH[I]*SORTIERATE*(100/URATE);
      IF NEWMAN > AIRMAN.POWER[I] THEN
        POWER[I] := NEWMAN
      ELSE
        POWER[I] := AIRMAN.POWER[I];
      IF POWER[I] < AIRMAN.MINCREW[I] THEN
        POWER[I] := AIRMAN.MINCREW[I];
      TOTPOW := TOTPOW + POWER[I];
    END; (*FOR*)
  END; (*BOOST*)

```

```

(*****
(* PROCEDURE : REPORT
(*
(* PURPOSE : Displays the manpower needed for each AFSC to
(*   meet the initial constraints.
(*
(* INPUT VARIABLES : None
(*
(* OUTPUT VARIABLES : None
(*
(* LOCAL VARIABLES :
(*   I - counter
(*****
PROCEDURE REPORT;

```

```

VAR
  I : INTEGER;

BEGIN (*REPORT*)
  CLRSCR;
  IF (READY1 = FALSE) OR (READY2 = FALSE) THEN
    WRITELN('Please select a weapon system and initial parameters first!')
  ELSE
    BEGIN (*ELSE*)
      ACCUMULATE(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH);

```

```

GETPOS(AIRMAN);
BOOST(POWER,TOTPOW);
FOR I:= 1 TO COUNT DO
  BEGIN (*FOR*)
    IF POWER[I]>AIRMAN.MINCREW[I] THEN
      WRITE(AIRMAN.AFS[I],' ',POWER[I]:6:3)
    ELSE
      WRITE(AIRMAN.AFS[I],' ',POWER[I]:6:3,' **DRIVEN BY CREW**');
    IF AIRMAN.FLAG[I] = TRUE THEN
      WRITELN(' *NO SOLUTION-ADDITIONAL MEN WILL HAVE LITTLE OR
NO EFFECT')
    ELSE
      WRITELN;
    END; (*FOR*)
    WRITELN('-----');
    WRITELN('TOTAL ',TOTPOW:6:3);
    END; (*ELSE*)
  WRITELN;
  WRITE('Press ENTER to return to the menu...');
  READLN;
END; (*REPORT*)

```

```

(*****)
(* PROCEDURE : HIDRIVER
(*
(* PURPOSE : To find and display the tasks of the AFSCs
(*   which require less manning than the minimum allowable.
(*
(* INPUT VARIABLES : None
(*
(* OUTPUT VARIABLES : None
(*
(* LOCAL VARIABLES :
(*   HI - records whether a HIDRIVER report is desired
(*   I - counter
(*   INPUT1 - input file
(*   AFSC, TASK, TIME, LAMBDA, CREW - information being read
(*   from the input file
(*   OLD - temporarily holds the current AFSC
(*   BLANK1, BLANK2 - hold spaces so the file can be read
(*****)
PROCEDURE HIDRIVER;

```

```

VAR
  I : INTEGER;

```



```

INPUT1 : TEXT;
AFS : STRING[6];
OLD : STRING[6];
BLANK1 : STRING[1];
TASK : STRING[8];
BLANK2 : STRING[1];
TIME : DOUBLE;
CREW : INTEGER;
LAMBDA : DOUBLE;
F : TEXT;
OUT : CHAR;
CHDEST : CHAR;
DEST : STRING;

BEGIN (*HIDRIVER*)
  CLRSCR;
  IF (READY1 = FALSE) OR (READY2 = FALSE) THEN
    WRITELN('Please select a weapon system and initial parameters first!')
  ELSE
    BEGIN (*ELSE*)
      WRITELN(' Hi Driver Report');
      WRITELN(' *****');
      ACCUMULATE(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH);
      GETPOS(AIRMAN);
      COUNT:=0;
      ASSIGN(INPUT1, NEWSET);
      RESET(INPUT1);
      WHILE NOT EOF(INPUT1) DO
        BEGIN (*WHILE*)
          OLD := AFS;
          READLN(INPUT1,AFS,BLANK1,TASK,BLANK2,TIME,CREW,LAMBDA);
          IF AFS <> OLD THEN
            COUNT:=COUNT+1;
          IF
(CREW>(AIRMAN.POWER[COUNT]+1))AND(CREW=AIRMAN.MINCREW[COUNT])
THEN
            WRITELN(AIRMAN.AFS[COUNT],' ',TASK);
          END; (*WHILE*)
        CLOSE(INPUT1);
        WRITELN;
        WRITELN('Would you like this to be sent to an output file? (y/n)');
        READLN(OUT);
        IF (OUT = 'Y') OR (OUT = 'y') THEN
          BEGIN (*IF*)
            WRITELN('The default destination is: c:\qman\hidriv.out.');
```

```

        WRITELN('Would you like to change the destination? (y,n)');
        READLN(CHDEST);
        IF (CHDEST = 'Y') OR (CHDEST = 'y') THEN
            BEGIN (*IF*)
                WRITE('The new destination is: ');
                READLN(DEST);
            END
        ELSE
            DEST := 'C:\QMAN\HIDRIV.OUT';
            ASSIGN(F, DEST);
            REWRITE(F);
            COUNT := 0;
            ASSIGN(INPUT1, NEWSET);
            RESET(INPUT1);
            WHILE NOT EOF(INPUT1) DO
                BEGIN (*WHILE*)
                    OLD := AFS;
                    READLN(INPUT1, AFS, BLANK1, TASK, BLANK2, TIME, CREW, LAMBDA);
                    IF AFS <> OLD THEN
                        COUNT := COUNT + 1;
                    IF
(CREW > (AIRMAN.POWER[COUNT] + 1)) AND (CREW = AIRMAN.MINCREW[COUNT])
THEN
                        WRITELN(F, AIRMAN.AFS[COUNT], ' ', TASK);
                    END; (*WHILE*)
                CLOSE(INPUT1);
                CLOSE(F);
            END; (*IF*)
        END; (*IF*)
        WRITELN;
        WRITE('Press ENTER to return to the menu...');
        READLN;
    END; (*HIDRIVER*)

```

```

(*****)
(* PROCEDURE : INDIVIDUAL
(*
(* PURPOSE : To calculate and display manpower for each AFSC
(*   from a reliability of 10% to 300%.
(*
(* INPUT VARIABLES : None
(*
(* OUTPUT VARIABLES : None
(*
(* LOCAL VARIABLES :

```

```

(* I, J - counters
(* Q - counter used calculate reliability
(* POS - required manpower
(* TEMP - mean arrival rate of planes
(* PEOPLE - number of crews needed for an AFSC cluster
(* INDIV - records whether an INDIVIDUAL report is desired
(* TEMPFLAG - warning that additional men will have little or
(* no effect on downtime
(*****))

```

PROCEDURE INDIVIDUAL;

VAR

```

I : INTEGER;
J : INTEGER;
Q : INTEGER;
POS : DOUBLE;
TEMP : DOUBLE;
PEOPLE : DOUBLE;
TEMPFLAG : BOOLEAN;
AF : INTEGER;
F : TEXT;
OUT : CHAR;
CHDEST : CHAR;
DEST : STRING;

```

BEGIN (*INDIVIDUAL*)

CLRSCR;

IF (READY1 = FALSE) OR (READY2 = FALSE) THEN

WRITELN('Please select a weapon system and initial parameters first!')

ELSE

BEGIN (*ELSE*)

ACCUMULATE(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH);

FOR I := 1 TO COUNT DO

WRITELN('I,' ,AIRMAN.AFS[I]);

WRITELN;

WRITE('Choose an AFSC to look at reliability: ');

READLN(AF);

WRITELN(AIRMAN.AFS[AF]);

FOR Q := 1 TO 30 DO

BEGIN (*FOR2*)

POS := 0;

FOR J := 1 TO WIDTH DO

BEGIN (*FOR3*)

TEMP := SORTIERATE*CLUSDEMAND[AF,J]*(Q/10);

IF ((CLUSTIME[AF,J]<0)AND(CCLUSDEMAND[AF,J]>0.00001))THEN

```

        BEGIN (*IF*)
            GETPEOPLE(PEOPLE,TEMPFLAG,TEMP,CLUSTIME[AF,J]);
            PEOPLE := PEOPLE*J;
            POS:=POS+PEOPLE;
        END; (*IF*)
    END; (*FOR3*)
    IF AIRMAN.MINCREW[AF]>POS THEN POS:=AIRMAN.MINCREW[AF];
    Writeln(ROUND((Q/10)*100),' ',ROUND(POS));
END; (*FOR2*)
Writeln;
Writeln('Would you like this to be sent to an output file? (y/n)');
Readln(OUT);
IF (OUT = 'Y') OR (OUT = 'y') THEN
    BEGIN (*IF*)
        Writeln('The default destination is: c:\qman\afsrel.out. ');
        Writeln('Would you like to change the destination? (y,n)');
        Readln(CHDEST);
        IF (CHDEST = 'Y') OR (CHDEST = 'y') THEN
            BEGIN (*IF*)
                WRITE('The new destination is: ');
                Readln(DEST);
            END
        ELSE
            DEST := 'C:\QMAN\AFSREL.OUT';
        ASSIGN(F, DEST);
        REWRITE(F);
        Writeln(F, AIRMAN.AFS[AF]);
        FOR Q := 1 TO 30 DO
            BEGIN (*FOR2*)
                POS := 0;
                FOR J := 1 TO WIDTH DO
                    BEGIN (*FOR3*)
                        TEMP := SORTIERATE*CLUSDEMAND[AF,J]*(Q/10);
                        IF ((CLUSTIME[AF,J]<>0)AND(CLUSDEMAND[AF,J]>0.00001))THEN
                            BEGIN (*IF*)
                                GETPEOPLE(PEOPLE,TEMPFLAG,TEMP,CLUSTIME[AF,J]);
                                PEOPLE := PEOPLE*J;
                                POS:=POS+PEOPLE;
                            END; (*IF*)
                        END; (*FOR3*)
                    END; (*FOR2*)
                IF AIRMAN.MINCREW[AF]>POS THEN POS:=AIRMAN.MINCREW[AF];
                Writeln(F, ROUND((Q/10)*100),' ',ROUND(POS));
            END; (*FOR2*)
        CLOSE(F);
    END; (*IF*)

```

```

    END; (*ELSE*)
    WRITELN;
    WRITE('Press ENTER to return to the menu...');
    READLN;
END; (*INDIVIDUAL*)

(*****)
PROCEDURE UOUT;

VAR
    I,Q : INTEGER;
    NEWMAN : DOUBLE;
    TEMP : DOUBLE;
    OUT : CHAR;
    F : TEXT;
    CHDEST : CHAR;
    DEST : STRING;

BEGIN (*UOUT*)
    CLRSCR;
    IF (READY1 = FALSE) OR (READY2 = FALSE) THEN
        WRITELN('Please select a weapon system and initial parameters first!')
    ELSE
        BEGIN (*ELSE*)
            ACCUMULATE(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH);
            GETPOS(AIRMAN);
            WRITELN(' Utilization Report');
            WRITELN(' *****');
            WRITELN('AFS   60%   70%   80%   90%   100%');
            FOR I := 1 TO COUNT DO
                BEGIN (*FOR1*)
                    WRITE(AIRMAN.AFS[I]);
                    TEMP := 0;
                    FOR Q := 6 TO 10 DO
                        BEGIN (*FOR2*)
                            NEWMAN := AIRMAN.MANH[I]*SORTIERATE*(10/Q);
                            IF NEWMAN > AIRMAN.POWER[I] THEN
                                TEMP := NEWMAN
                            ELSE
                                TEMP := AIRMAN.POWER[I];
                            IF AIRMAN.MINCREW[I] > TEMP THEN
                                TEMP := AIRMAN.MINCREW[I];
                            WRITE(' ',TEMP:6:3);
                        END; (*FOR2*)
                    WRITELN;
                END;
            END;
        END;
    END;

```

```

    END; (*FOR1*)
    WRITELN;
    WRITELN('Would you like this to be sent to an output file? (y/n)');
    READLN(OUT);
    IF (OUT = 'Y') OR (OUT = 'y') THEN
        BEGIN (*IF*)
            WRITELN('The default destination is: c:\qman\util.out. ');
            WRITELN('Would you like to change the destination? (y,n)');
            READLN(CHDEST);
            IF (CHDEST = 'Y') OR (CHDEST = 'y') THEN
                BEGIN (*IF*)
                    WRITE('The new destination is: ');
                    READLN(DEST);
                END
            ELSE
                DEST := 'C:\QMAN\UTIL.OUT';
            ASSIGN(F, DEST);
            REWRITE(F);
            FOR I := 1 TO COUNT DO
                BEGIN (*FOR1*)
                    WRITE(F, AIRMAN.AFS[I]);
                    TEMP := 0;
                    FOR Q := 6 TO 10 DO
                        BEGIN (*FOR2*)
                            NEWMAN := AIRMAN.MANH[I]*SORTIERATE*(10/Q);
                            IF NEWMAN > AIRMAN.POWER[I] THEN
                                TEMP := NEWMAN
                            ELSE
                                TEMP := AIRMAN.POWER[I];
                            IF AIRMAN.MINCREW[I] > TEMP THEN
                                TEMP := AIRMAN.MINCREW[I];
                            WRITE(F, ', ', TEMP:6:3);
                        END; (*FOR2*)
                    WRITELN(F);
                END; (*FOR1*)
            CLOSE(F);
        END; (*IF*)
    END; (*ELSE*)
    WRITELN;
    WRITE('Press ENTER to return to the menu...');
    READLN;
END; (*UOUT*)

(*****)
PROCEDURE WEAVOUT;

```

```

VAR
  I : INTEGER;
  UTIL : ARRAYTYPE;

BEGIN (*WEAVOUT*)
  CLRSCR;
  WRITELN('  AFSC MANH POW UTIL');
  FOR I := 1 TO COUNT DO
    IF POWER[I]>AIRMAN.MINCREW[I] THEN
      BEGIN
        UTIL[I] := (AIRMAN.MANH[I]*SORTIERATE)/POWER[I];
        WRITELN('(',I,')
',AIRMAN.AFS[I],AIRMAN.MANH[I]:6:3,POWER[I]:7:3,UTIL[I]:6:3);
      END
    ELSE
      BEGIN
        UTIL[I] := (AIRMAN.MANH[I]*SORTIERATE)/AIRMAN.MINCREW[I];
        WRITELN('(',I,')
',AIRMAN.AFS[I],AIRMAN.MANH[I]:6:3,AIRMAN.MINCREW[I]:7:3,UTIL[I]:6:3);
      END;
    END; (*WEAVOUT*)

```

```

(*****
PROCEDURE AFSC;

```

```

VAR
  NUM : INTEGER;
  COMBO : CHAR;
  A : INTEGER;
  B : INTEGER;
  C : INTEGER;
  D : INTEGER;
  WORK : DOUBLE;
  J : INTEGER;
  TEMPFLAG : BOOLEAN;
  TEMP : DOUBLE;
  PEOPLE : DOUBLE;
  NEWMAN : DOUBLE;
  UTIL : ARRAYTYPE;
  POW : DOUBLE;

```

```

BEGIN (*AFSC*)
  CLRSCR;
  IF (READY1 = FALSE) OR (READY2 = FALSE) THEN

```

```

        WRITELN('Please select a weapon system and initial parameters first!')
ELSE
    BEGIN (*ELSE*)
        ACCUMULATE(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH);
        GETPOS(AIRMAN);
        BOOST(POWER,TOTPOW);
        WEAVOUT;
        WRITE('Do you want to combine 2 or 3 AFS"s? ');
        READLN(NUM);
        WRITE('Enter the # of the first AFS to combine: ');
        READLN(A);
        WRITE('Enter the # AFS to combine with ', A, ': ');
        READLN(B);
        IF NUM = 3 THEN
            BEGIN (*IF*)
                WRITE('Enter the # AFS to combine with ', A, ' and ', B, ': ');
                READLN(D);
            END; (*IF*)
        WRITE('At what % of their previous rate will they be working? ');
        READLN(WORK);
        POW := 0;
        C := COUNT + 1;
        AIRMAN.AFS[C] := 'NEW';
        IF NUM <> 3 THEN
            BEGIN (*IF*)
                IF AIRMAN.MINCREW[A] < AIRMAN.MINCREW[B] THEN
                    AIRMAN.MINCREW[C] := AIRMAN.MINCREW[A]
                ELSE
                    AIRMAN.MINCREW[C] := AIRMAN.MINCREW[B];
                AIRMAN.MANH[C] := AIRMAN.MANH[A] + AIRMAN.MANH[B];
            END (*IF*)
        ELSE
            BEGIN (*ELSE*)
                IF (AIRMAN.MINCREW[A] < AIRMAN.MINCREW[B]) AND
                    (AIRMAN.MINCREW[A] < AIRMAN.MINCREW[D]) THEN
                    AIRMAN.MINCREW[C] := AIRMAN.MINCREW[A]
                ELSE
                    IF AIRMAN.MINCREW[B] < AIRMAN.MINCREW[D] THEN
                        AIRMAN.MINCREW[C] := AIRMAN.MINCREW[B]
                    ELSE
                        AIRMAN.MINCREW[C] := AIRMAN.MINCREW[D];
                    AIRMAN.MANH[C] := AIRMAN.MANH[A] + AIRMAN.MANH[B] +
AIRMAN.MANH[D];
                END; (*ELSE*)
            END
        FOR J := 1 TO WIDTH DO

```



```

BEGIN (*FOR*)
  IF NUM <> 3 THEN
    BEGIN (*IF*)
      CLUSDEMAND[C,J] := CLUSDEMAND[A,J] + CLUSDEMAND[B,J];
      IF CLUSDEMAND[C,J] > 0 THEN

CLUSTIME[C,J]:=((CTIME[A,J]+CTIME[B,J])/CLUSDEMAND[C,J])*100/WORK;
      END (*IF*)
    ELSE
      BEGIN (*ELSE*)

CLUSDEMAND[C,J]:=CLUSDEMAND[A,J]+CLUSDEMAND[B,J]+CLUSDEMAND[D,J];
      IF CLUSDEMAND[C,J] > 0 THEN
        CLUSTIME[C,J]
:=((CTIME[A,J]+CTIME[B,J]+CTIME[D,J])/CLUSDEMAND[C,J])*100/WORK;
        END; (*ELSE*)
      TEMPFLAG := FALSE;
      TEMP := SORTIERATE*CLUSDEMAND[C,J];
      IF ((CLUSTIME[C,J]<>0)AND(CLUSDEMAND[C,J]>0.00001))THEN
        BEGIN (*IF*)
          GETPEOPLE(PEOPLE,TEMPFLAG,TEMP,CLUSTIME[C,J]);
          PEOPLE := PEOPLE*J;
          POW := POW + PEOPLE;
        END; (*IF*)
      END; (*FOR*)
      GETPOS(AIRMAN);
      BOOST(POWER, TOTPOW);
      WEAVOUT;
      WRITELN;
      NEWMAN := AIRMAN.MANH[C]*SORTIERATE*(100/URATE);
      IF NEWMAN > POW THEN
        POW := NEWMAN;
      IF POW > AIRMAN.MINCREW[C] THEN
        BEGIN
          UTIL[C] := (AIRMAN.MANH[C]*SORTIERATE)/POW;
          IF NUM <> 3 THEN

WRITELN('(',A,',',B,')',AIRMAN.AFS[C],AIRMAN.MANH[C]:6:3,POW:7:3,UTIL[C]:6:3)
          ELSE

WRITELN('(',A,',',B,',',D,')',AIRMAN.AFS[C],AIRMAN.MANH[C]:6:3,POW:7:3,UTIL[C]:6:3)
          END
        ELSE
          BEGIN
            UTIL[C] := (AIRMAN.MANH[C]*SORTIERATE)/AIRMAN.MINCREW[C];

```

```

        IF NUM <> 3 THEN

WRITELN('(',A,',',B,')',AIRMAN.AFS[C],AIRMAN.MANH[C]:6:3,AIRMAN.MINCREW[C]:7:
3,UTIL[C]:6:3)
        ELSE

WRITELN('(',A,',',B,',',D,')',AIRMAN.AFS[C],AIRMAN.MANH[C]:6:3,AIRMAN.MINCREW[
C]:7:3,UTIL[C]:6:3)
        END;
        END; (*ELSE*)
WRITELN;
WRITE('Press ENTER to return to the menu...');
READLN;
END; (*AFSC*)

(*****)
PROCEDURE ACCUM2(VAR AIRMAN : MAN;
        VAR CTIME : CLUSTYPE;
        VAR CLUSTIME : CLUSTYPE;
        VAR CLUSDEMAND : CLUSTYPE;
        VAR COUNT : INTEGER;
        VAR WIDTH : INTEGER;
        OUT : CHAR);

VAR
        INPUT1 : TEXT;
        AFSC : STRING[6];
        OLD : STRING[6];
        TASK : STRING[8];
        TIME : DOUBLE;
        CREW : INTEGER;
        LAMBDA : DOUBLE;
        BLANK1 : STRING[1];
        BLANK2 : STRING[1];
        TEMP : STRING[2];
        I : INTEGER;
        J : INTEGER;
        K : INTEGER;
        L : INTEGER;
        F : TEXT;
        CHDEST : CHAR;
        DEST : STRING;

BEGIN (*ACCUM2*)
        IF (OUT = 'Y') OR (OUT = 'y') THEN

```

```

BEGIN (*IF*)
  WRITELN('The default destination is: c:\qman\eqrel.out. ');
  WRITELN('Would you like to change the destination? (y,n)');
  READLN(CHDEST);
  IF (CHDEST = 'Y') OR (CHDEST = 'y') THEN
    BEGIN (*IF*)
      WRITE('The new destination is: ');
      READLN(DEST);
    END
  ELSE
    DEST := 'C:\QMAN\EQREL.OUT';
  END; (*IF*)
ASSIGN(INPUT1, NEWSET);
ASSIGN(F, DEST);
REWRITE(F);
FOR J := 1 TO 15 DO
  BEGIN (*FOR*)
    FOR K := 1 TO MAXAFSCLENGTH DO
      BEGIN (*FOR1*)
        AIRMAN.MINCREW[K] := 0;
        AIRMAN.MANH[K] := 0;
        FOR L := 1 TO MAXCREW DO
          BEGIN (*FOR2*)
            CTIME[K,L]:=0;
            CLUSTIME[K,L]:=0;
            CLUSDEMAND[K,L]:=0;
          END; (*FOR2*)
        END; (*FOR1*)
      COUNT := 0;
      WIDTH := 0;
      RESET(INPUT1);
      WHILE NOT EOF(INPUT1) DO
        BEGIN (*WHILE*)
          OLD:=AFSC;
          READLN(INPUT1,AFSC,BLANK1,TASK,BLANK2,TIME,CREW,LAMBDA);
          TEMP := COPY(TASK,4,2);
          IF TEMP = EQ THEN
            LAMBDA := LAMBDA *J/5;
          IF OLD<>AFSC THEN
            COUNT:=COUNT+1;
            AIRMAN.AFS[COUNT]:=AFSC;
            IF AIRMAN.MINCREW[COUNT]<CREW THEN
              AIRMAN.MINCREW[COUNT]:=CREW;
            CTIME[COUNT,CREW]:=CTIME[COUNT,CREW]+(LAMBDA*TIME);
            CLUSDEMAND[COUNT,CREW]:=CLUSDEMAND[COUNT,CREW]+LAMBDA;

```

```

        IF CREW > WIDTH THEN
            WIDTH:=CREW;

AIRMAN.MANH[COUNT]:=AIRMAN.MANH[COUNT]+(CREW*TIME*LAMBDA);
        END; (*WHILE*)
        FOR K:= 1 TO MAXAFSCLENGTH DO
            FOR L:= 1 TO MAXCREW DO
                IF CLUSDEMAND[K,L]>0 THEN
                    CLUSTIME[K,L]:=CTIME[K,L]/CLUSDEMAND[K,L];
                BOOST(POWER,TOTPOW);
                IF (OUT = 'Y') OR (OUT = 'y') THEN
                    WRITELN(F, ROUND((J/5)*100),' ',TOTPOW:6:3)
                ELSE
                    WRITELN(ROUND((J/5)*100),' ',TOTPOW:6:3);
                END; (*FOR*)
            CLOSE(F);
            CLOSE(INPUT1);
        END; (*ACCUM2*)

```

```

(*****)
PROCEDURE REPORT2(VAR EQ : EQTYPE);

```

```

VAR
    I : INTEGER;
    J : INTEGER;
    OUT : CHAR;

BEGIN (*REPORT2*)
    J := 1;
    WRITELN('Equipages: ');
    FOR I := 1 TO COUNT2 DO
        BEGIN (*FOR*)
            J := J + 1;
            WRITE(E[I],' ');
            IF J = 10 THEN
                BEGIN (*IF*)
                    WRITELN;
                    J := 1;
                END; (*IF*)
            END; (*FOR*)
        WRITELN;
        WRITELN;
        WRITELN('Choose an equipage to look at reliability: ');
        READLN(EQ);
        WRITELN;
    
```

```

WRITELN('Equipage - ',EQ);
WRITELN('% Total Manpower');
OUT := 'N';
ACCUM2(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH,OUT);
END; (*REPORT2*)

(*****)
PROCEDURE EQUIPAGE(VAR COUNT2 : INTEGER;
                   VAR E : EQUIPTYPE);

VAR
  INPUT1 : TEXT;
  AFSC : STRING[6];
  TASK : STRING[8];
  TIME : DOUBLE;
  CREW : INTEGER;
  LAMBDA : DOUBLE;
  BLANK1 : STRING[1];
  BLANK2 : STRING[1];
  DONE : BOOLEAN;
  X : STRING[10];
  T : ARRAY[1..2] OF STRING[1];
  FLAG : ARRAY[1..2] OF BOOLEAN;
  I : INTEGER;
  OUT : CHAR;

BEGIN (*EQUIPAGE*)
  CLRSCR;
  IF (READY1 = FALSE) OR (READY2 = FALSE) THEN
    WRITELN('Please select a weapon system and initial parameters first!')
  ELSE
    BEGIN (*ELSE*)
      ASSIGN(INPUT1, NEWSET);
      RESET(INPUT1);
      COUNT2 := 0;
      WHILE NOT EOF(INPUT1) DO
        BEGIN (*WHILE1*)
          READLN(INPUT1,AFSC,BLANK1,TASK,BLANK2,TIME,CREW,LAMBDA);
          FOR I := 1 TO 2 DO
            BEGIN (*FOR*)
              T[I] := COPY(TASK,I+3,1);
              IF (T[I]='1') OR (T[I]='2') OR (T[I]='3') OR (T[I]='4') OR
                 (T[I]='5') OR (T[I]='6') OR (T[I]='7') OR (T[I]='8')
                 OR (T[I]='9') OR (T[I]='0') THEN
                FLAG[I] := FALSE
            END
          END
        END
      END
    END
  END
END

```

```

ELSE
    FLAG[I] := TRUE;
END; (*FOR*)
IF (FLAG[1] = FALSE) AND (FLAG[2] = FALSE) THEN
    BEGIN (*IF1*)
        COUNT2 := COUNT2 + 1;
        E[COUNT2] := COPY(TASK,4,2);
        IF COUNT2 > 1 THEN
            BEGIN (*IF2*)
                DONE := FALSE;
                I := 1;
                REPEAT
                    IF E[COUNT2] = E[I] THEN
                        BEGIN (*IF3*)
                            DONE := TRUE;
                            COUNT2 := COUNT2 - 1;
                        END (*IF3*)
                    ELSE
                        I := I + 1;
                        IF I = COUNT2 THEN
                            DONE := TRUE;
                        UNTIL (DONE = TRUE);
                    END; (*IF2*)
                END; (*IF1*)
            END; (*WHILE1*)
        REPORT2(EQ);
        WRITELN;
        WRITELN('Would you like this to be sent to an output file? (y/n)');
        READLN(OUT);
        IF (OUT = 'Y') OR (OUT = 'y') THEN
            ACCUM2(AIRMAN,CTIME,CLUSTIME,CLUSDEMAND,COUNT,WIDTH,OUT);
        END; (*ELSE*)
        WRITELN;
        WRITE('Press ENTER to return to the menu...');
        READLN;
    END; (*EQUIPAGE*)

```

```

(*****

```

```

BEGIN (*MAIN*)
    INTRO;
    READY1 := FALSE;
    READY2 := FALSE;
    SELECT := 0;
    WHILE SELECT <> 9 DO
        BEGIN (*WHILE*)

```

```
MENU(SELECT);  
CASE SELECT OF  
  1 : WEAPON(NEWSET,READY1);  
  2 : GETINFO(SORTIERATE,WINDOW,ALPHA,URATE,READY2);  
  3 : REPORT;  
  4 : UOUT;  
  5 : HIDRIVER;  
  6 : INDIVIDUAL;  
  7 : EQUIPAGE(COUNT2,E);  
  8 : AFSC;  
END; (*CASE*)  
END; (*WHILE*)  
END. (*MAIN*)
```

APPENDIX B: QMAN TO LCOM COMPARISON (F-22)

Comparison of QMAN and LCOM Manpower Estimates 24 F-22 aircraft flying three 2-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
326X6	9	8	7	6	6	8
327X7	3	2	2	2	2	2
328X8	2	2	2	2	2	2
404X1	2	2	2	2	2	2
423X0	3	3	3	3	3	3
423X1	3	3	3	3	3	3
423X2	2	2	2	2	2	2
423X3	5	5	4	4	4	5
423X4	3	3	3	3	3	3
426X2	6	5	4	4	4	5
427X1	1	1	1	1	1	1
427X2	2	2	2	2	2	2
427X5	4	3	3	3	3	4
431F1 ^a	8	7	6	5	5	7
431R1 ^a	2	2	2	2	2	2
462LO ^a	6	6	6	6	6	6
462X0 ^a	3	3	3	3	3	3
Mean	3.76	3.47	3.24	3.12	3.12	3.53
SD	2.28	2.00	1.68	1.45	1.45	2.00

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 24 F-22 aircraft flying two 2-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
326X6	6	6	5	4	4	4
327X7	2	2	2	2	2	2
328X8	2	2	2	2	2	2
404X1	2	2	2	2	2	2
423X0	3	3	3	3	3	3
423X1	3	3	3	3	3	3
423X2	2	2	2	2	2	2
423X3	4	3	3	3	2	3
423X4	3	3	3	3	3	3
426X2	4	4	4	4	4	4
427X1	1	1	1	1	1	1
427X2	2	2	2	2	2	2
427X5	3	2	2	2	2	2
431F1 ^a	5	5	4	4	3	4
431R1 ^a	2	2	2	2	2	2
462LO ^a	4	3	3	3	3	3
462X0 ^a	3	3	3	3	3	3
Mean	3.00	2.82	2.71	2.65	2.53	2.65
SD	1.27	1.24	0.99	0.86	0.80	0.86

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 48 F-22 aircraft flying two 1.5-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
326X6	12	11	9	8	8	9
327X7	3	3	3	2	2	3
328X8	3	3	2	2	2	2
404X1	2	2	2	2	2	2
423X0	4	4	3	3	3	3
423X1	3	3	3	3	3	3
423X2	2	2	2	2	2	2
423X3	7	6	5	5	4	5
423X4	3	3	3	3	3	3
426X2	7	6	6	5	5	6
427X1	1	1	1	1	1	1
427X2	2	2	2	2	2	2
427X5	5	4	4	3	3	4
431F1 ^a	10	9	8	7	6	8
431R1 ^a	2	2	2	2	2	2
462LO ^a	7	6	6	6	6	6
462X0 ^a	3	3	3	3	3	3
Mean	4.47	4.12	3.76	3.47	3.35	3.76
SD	3.10	2.69	2.28	2.00	1.87	2.28

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 72 F-22 aircraft flying three 1.5-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
326X6	27	23	20	18	18	20
327X7	7	6	6	6	6	7
328X8	6	5	5	4	4	5
404X1	2	2	2	2	2	2
423X0	9	8	8	8	8	8
423X1	3	3	3	3	3	3
423X2	2	2	2	2	2	2
423X3	15	13	11	10	10	11
423X4	4	4	4	4	4	5
426X2	16	15	12	12	12	14
427X1	1	1	1	1	1	1
427X2	2	2	2	2	2	2
427X5	10	9	8	7	6	8
431F1 ^a	23	20	17	15	15	17
431R1 ^a	2	2	2	2	2	2
462LO ^a	15	13	12	10	9	12
462X0 ^a	3	3	3	3	3	3
Mean	8.65	7.71	6.94	6.41	6.29	7.18
SD	7.94	6.80	5.70	5.06	5.02	5.79

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 72 F-22 aircraft flying three 2-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
326X6	27	23	20	18	18	23
327X7	7	6	6	6	6	7
328X8	6	5	5	5	5	5
404X1	2	2	2	2	2	2
423X0	9	8	8	8	8	8
423X1	3	3	3	3	3	3
423X2	2	2	2	2	2	4
423X3	15	13	11	10	10	15
423X4	4	4	4	4	4	5
426X2	16	14	12	12	12	14
427X1	1	1	1	1	1	1
427X2	2	2	2	2	2	2
427X5	10	9	8	7	6	9
431F1 ^a	23	20	17	15	15	20
431R1 ^a	2	2	2	2	2	3
462LO ^a	15	13	12	12	12	15
462X0 ^a	3	3	3	3	3	3
Mean	8.65	7.65	6.94	6.59	6.53	8.18
SD	7.94	6.74	5.70	5.15	5.15	6.78

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

APPENDIX C: QMAN TO LCOM COMPARISON (F-15)

Comparison of QMAN and LCOM Manpower Estimates 24 F-15 aircraft flying three 2-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
324X0	1	1	1	1	1	1
326E8 ^a	2	2	2	2	2	2
326X6	6	6	6	6	6	6
326X7	2	2	2	2	2	2
326X8	5	4	4	3	3	4
423X0	3	2	2	2	2	4
423X1	2	2	2	2	2	2
423X4	2	2	2	2	2	4
426X2	21	21	21	21	21	21
427X0	2	2	2	2	2	2
427X5	4	3	3	3	2	3
431E1 ^a	13	12	10	9	8	13
431F1 ^a	5	5	5	5	5	5
431R1 ^a	5	5	5	5	5	5
462F0 ^a	2	2	2	2	2	2
462L0 ^a	13	11	10	9	9	12
462X0 ^a	6	5	4	4	4	6
Mean	5.53	5.12	4.88	4.71	4.59	5.53
SD	5.34	5.16	4.95	4.84	4.82	5.21

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 48 F-15 aircraft flying two 1.5-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
324X0	1	1	1	1	1	1
326E8 ^a	2	2	2	2	2	2
326X6	6	6	6	6	6	5
326X7	3	3	2	2	2	2
326X8	6	5	5	4	4	4
423X0	4	3	3	3	2	3
423X1	2	2	2	2	2	2
423X4	3	3	2	2	2	2
426X2	18	18	18	18	18	15
427X0	2	2	2	2	2	2
427X5	5	4	4	3	3	5
431E1 ^a	18	15	13	12	11	12
431F1 ^a	5	5	5	5	5	5
431R1 ^a	5	5	5	5	5	5
462F0 ^a	2	2	2	2	2	2
462L0 ^a	17	15	13	12	12	12
462X0 ^a	7	6	6	6	6	6
Mean	6.24	5.71	5.35	5.12	5.00	5.00
SD	5.72	5.17	4.82	4.66	4.61	4.14

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 48 F-15 aircraft flying three 2-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
324X0	1	1	1	1	1	1
326E8 ^a	3	3	2	2	2	2
326X6	9	8	8	8	8	8
326X7	4	4	3	3	3	4
326X8	9	8	7	6	6	8
423X0	5	4	4	4	4	5
423X1	2	2	2	2	2	3
423X4	6	6	6	6	6	6
426X2	29	29	29	29	29	28
427X0	2	2	2	2	2	2
427X5	7	6	5	5	4	6
431E1 ^a	26	23	20	18	16	22
431F1 ^a	5	5	5	5	5	5
431R1 ^a	5	5	5	5	5	5
462F0 ^a	3	3	2	2	2	3
462L0 ^a	25	22	19	18	18	21
462X0 ^a	11	10	10	10	10	11
Mean	8.94	8.29	7.65	7.41	7.24	8.24
SD	8.91	8.27	7.80	7.53	7.40	7.89

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 72 F-15 aircraft flying three 1.5-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
324X0	1	1	1	1	1	1
326E8 ^a	4	4	4	4	4	4
326X6	13	11	10	9	9	10
326X7	6	6	6	6	6	6
326X8	13	11	10	9	8	10
423X0	7	6	6	5	5	6
423X1	3	2	2	2	2	3
423X4	6	6	6	6	6	5
426X2	39	35	35	35	35	36
427X0	2	2	2	2	2	2
427X5	10	8	7	7	6	7
431E1 ^a	39	34	29	26	26	29
431F1 ^a	7	6	6	5	5	6
431R1 ^a	6	5	5	5	5	5
462F0 ^a	4	4	4	4	4	4
462L0 ^a	38	33	29	25	24	28
462X0 ^a	16	14	13	13	13	13
Mean	12.59	11.06	10.29	9.65	9.47	10.29
SD	13.08	11.46	10.41	9.72	9.65	10.43

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.

Comparison of QMAN and LCOM Manpower Estimates 72 F-15 aircraft flying three 2-hour sorties per aircraft per day (N=17)						
AFSC	QMAN-60% Max Ut	QMAN-70% Max Ut	QMAN-80% Max Ut	QMAN-90% Max Ut	QMAN-100% Max Ut	LCOM
324X0	1	1	1	1	1	1
326E8 ^a	4	4	4	4	4	4
326X6	13	11	10	9	9	11
326X7	6	6	6	6	6	6
326X8	13	11	10	9	8	11
423X0	7	6	6	5	5	6
423X1	3	2	2	2	2	3
423X4	6	6	6	6	6	6
426X2	39	38	38	38	38	38
427X0	2	2	2	2	2	2
427X5	10	8	7	7	6	8
431E1 ^a	39	34	29	26	26	33
431F1 ^a	7	6	6	5	5	6
431R1 ^a	6	6	6	6	6	5
462F0 ^a	4	4	4	4	4	4
462L0 ^a	38	33	29	25	24	33
462X0 ^a	16	13	13	13	13	13
Mean	12.59	11.24	10.53	9.88	9.71	11.18
SD	13.08	11.83	10.84	10.20	10.14	11.70

a = The letter other than X in the fourth position of these AFSs designates a subdivision of an AFS that is distinguished by its physical work location.